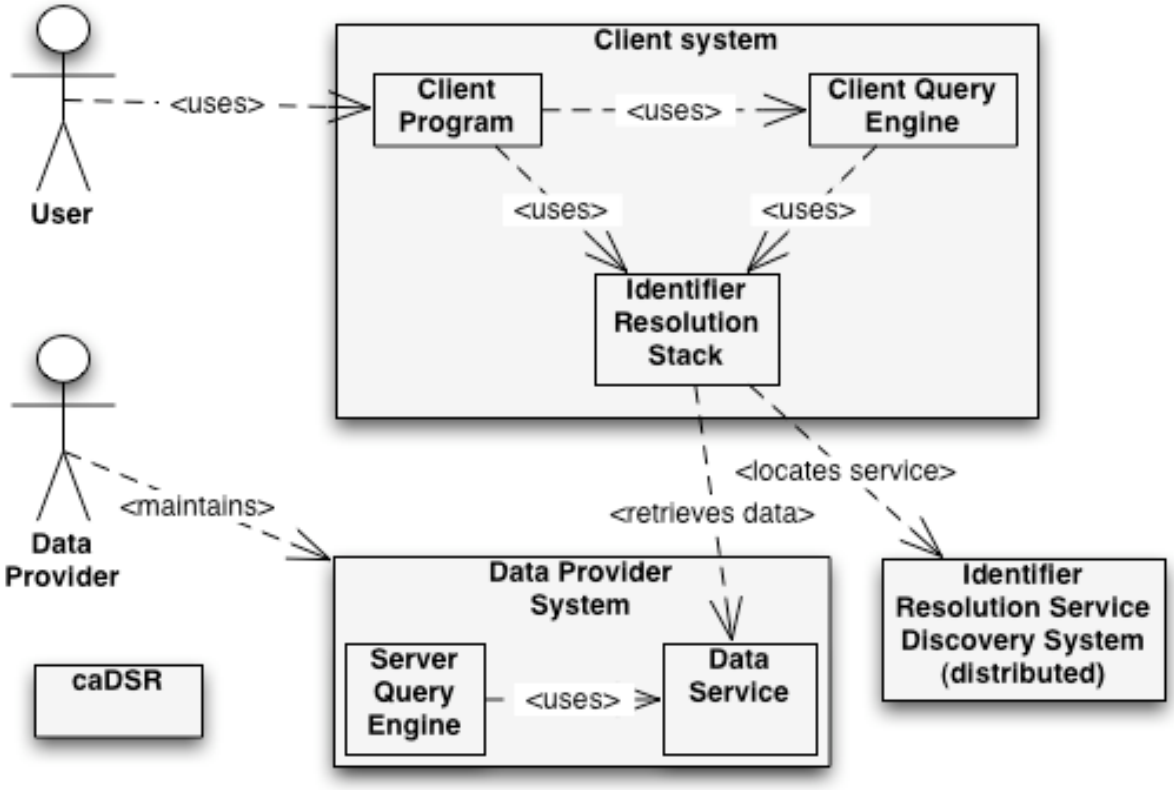
8.1 Design scope drawing



The following terms are used in the use cases and the diagram above:

User: party that wants to access caBIG data resources

Data Provider: party that wants to provide access to its data via caBIG

caDSR (Cancer Data Standards Repository): the caBIG object model repository for all caBIG data objects

Client Program: software used by user to access caBIG grid; this software uses various caBIG libraries and APIs to access grid resources

Query Engine: not part of identifier resolution, but an important client; caBIG software library that parses queries written in the standard caBIG query language and performs distributed grid queries at the request of a Client Program or other caBIG libraries. The Server Query Engine provides server side services related to distributed query.

Identifier Resolution Stack (Resolution Stack for short): the tool used on the client side to resolve data associated with LSIDs; it will abstract away the details of LSID resolution, providing some insulation from future changes in the LSID specification.

Data Service: the Data Provider maintains this service, which wraps the Data Provider's data resources in a way that makes them accessible to the grid

Identifier Resolution Service Discovery System (IRSDS): distributed system that allows the Resolution Stack to find the service that provides the data associated with a given identifier; not meant to provide arbitrary service discovery functionality

8.2 In/out list

This table indicates which topics are in or out of the scope of this specification and the identifiers SIG. Being out of scope for this specification does not mean out of scope for caBIG; instead, out of scope may simply indicate that topics may be more appropriately addressed by another SIG or group (as indicated).

Topic	In	Out	More Appropriate SIG
Retrieve data by LSID	In		
Service discovery by service attribute		Out	Common Query Language
Secure access to data		Out	Security

8.3 Actor-goal list

Actor	Task-level Goal	Priority
Client Program	Get digital data	
Client Program	Get relationships between biological objects and digital data	
Client Program	Get digital data associated with a biological object	
Client Program	Get specific version of digital data (by version number or by timestamp)	
Client Program	Manage cached data objects	
Data Provider	Provide identifiers for data objects	
Data Provider	Provide data relationship information	
Data Provider	Provide minimal caBIG metadata (provenance, etc.)	
Data Provider	Move service to new host	
Data Provider	Cluster service at multiple hosts	
Data Provider	Mirror existing service	

8.4 Usage narratives

These are intended to give a feel for the types of caBIG usage that will indirectly involve data identifiers. Typical users will not use identifiers directly; usually distributed queries will return identifiers, and software will then use these identifiers to retrieve the associated data. Providing grid identifiers will impose software contractual requirements on data providers, and the characteristics of these requirements may be inferred from typical usages as well.

I am not a typical caBIG user, so these narratives are somewhat imagined. Please correct me as necessary.

8.4.1 Repeating and evolving someone else's analysis

The typical caBIG user, let's call him Joe, will not use data identifiers directly. Joe has heard of Mary's gene expression study of the effect of a drug on expression levels in some kind of cancer tissue. He would like to repeat her calculations using a slightly different clustering method, so that he can be sure that he believes her results. So, he uses a grid query to obtain her data. Under the hood, the query engine uses the data's grid identifiers to fetch the data. He then feeds this data to some locally installed clustering algorithms. After confirming these results, he wants to learn more about the tissue samples used in her experiment, so that he can understand the staging and clinical details of the patients from which they were taken. Again, he uses a grid query, again avoiding actually using grid identifiers directly. Grid identifiers are used as foreign keys between the tissue bank and the clinical data about the patients. Upon finding some interesting details about the patients, he wants to find a list of tissue samples available from patients with those characteristics at a set of nearby institutions, so that he can order samples and repeat the experiment with variations interesting to him. The query engine performs a distributed join to return this data to him.

8.4.2 Refining a complex calculation

Emily performs a complex calculation with multiple steps involving multiple data sources. She caches some data locally for performance reasons. As she tweaks the calculation, she stores important intermediate results, so that she can track down any surprising changes in the final answer. This narrative should be fleshed out more.

8.4.3 21 CFR Part 11 Compliant organization

Electronic data records that fall under Part 11 have special requirements. An organization, call it XYZ Pharmaceuticals, is Part 11 compliant and uses caBIG resources in its clinical/translational research. Electronic records that are a part of regulated activities must have an appropriate audit trail, and XYZ must have policies in place for record retention and systems validation, and must be able to provide copies of records upon request.

Local storage scenario: XYZ stores a copy of any data objects that XYZ retrieves from the grid that are a part of regulated activities. In this way, XYZ does not have to depend on external organizations' Part 11 compliance, and can regulate their own activity to their own satisfaction. However, this means they will have to do some "cache management". For example, an XYZ researcher may want to know whether a local copy of a data object is the latest version.

Distributed storage scenario: if XYZ can be assured that an external data provider is Part 11 compliant, they may be able to leave data stored on the data provider's servers, and not store a copy locally. In this case, the data provider must provide record retention and audit trail. A given version of a data object must be retrievable later using the revision-specific identifier, and it must be byte-identical to the record originally retrieved by XYZ during research. This scenario would provide the ability to furnish copies of records as a natural byproduct. However, the data provider would have to be able to certify that it would

not discard old versions of data objects. Perhaps some data providers could get certified as being Part 11 compliant, so that other Part 11 compliant organizations would feel comfortable using them in this way.

8.5 Use cases

I have written these use cases in a (somewhat) casual style, because I felt that going into too much detail would be inappropriate at this stage.

Primary Actor is the protagonist of the use case. Scope is a list of the subsystems affected by the use case (shown in the design scope drawing). Level is either User Goal for a specific user goal, or Summary if I think it represents several user goals. The Actor-Goal list is a bit high-level, so some of the goals there are summaries here.

Use Case 1 Get data object associated with identifier

Primary Actor: Client Program on behalf of User
Scope: Client System, Data Provider System and IRSDS
Level: User Goal
Precondition: Client Program has identifier for desired data object
The Client Program requests (XML) data associated with the LSID from the Identifier Resolution Stack, which uses the IRSDS to find the Data Provider System and then asks the Data Service for the data object's XML. If the identifier contains a Revision, then the Data Service should return that version of the data object. If the identifier does not contain a Revision, then the Data Service should return the latest version of the data object.
If no data is available for the given identifier (because either a specific version or an entire object is unavailable), the Data Service should report this to the Resolution Stack, which then reports this to the Client Program. The Resolution Stack should also be able to timeout on its attempt to contact the Data Service, and report that the service is currently unavailable.

Use Case 2 Get data objects associated with a list of identifiers

Primary Actor: Client Program on behalf of User
Scope: Client System, Data Provider System and IRSDS
Level: User Goal
Precondition: Client Program has a list of identifiers for the desired data objects
Similar to Use Case 1 except that the Client Program presents a list of identifiers to the Resolution Stack, and the Resolution Stack returns a corresponding list of XML objects. In this case, the Data Provider and Resolution Stack will need to be able to provide notification of partial success/failure, in the event that data is available for some of the identifiers and unavailable for others.

Use Case 3 Get relationships between biological objects and digital data about those objects

Primary Actor: Client Program on behalf of User
Scope: Client System, Data Provider System and IRSDS
Level: User Goal
Note: This Use Case is a bit bloated, in that I have included thoughts about how LSIDs might be used to provide identifiers for biological objects.
Some identifiers will refer to biological objects, for example a specific patient or gene. These biological object identifiers will not refer to actual data, but to metadata relationships between the biological object and identifiers for digital data about the object in various forms.
Biological object identifiers are necessary because one may want to provide a foreign key for the entire patient, and not just for a particular representation of data about that patient. Because the data associated with a patient could have a fairly arbitrary and changing schema (to say nothing of even more

frequent data changes), the Data Service associates no actual data with the patient's biological object identifier.
The Client Program asks the Resolution Stack for the metadata associated with the biological object identifier. The Resolution Stack (after discovering and querying the Data Service) returns a list of relationships to identifiers for related digital data in various forms. The relationships also encode the type of data available at each of those identifiers.

Use Case 4 Manage cached data objects

Primary Actor: Client Program on behalf of User
Scope: Client System, Data Provider System and IRSDS
Level: User Goal
The Client Program has a local data store that contains copies of data objects (for performance or regulatory compliance purposes, for example) previously retrieved from other data providers on the grid. The User wants to use only the latest version available of a particular data object, but would like to use the locally cached version if possible. The Client Program passes the revisioned identifier to the Resolution Stack, which discovers the Data Service and requests the identifier for the latest version of the data object. The Resolution Stack compares this identifier's revision to the revision of the locally stored data object, reporting the result to the Client Program.
In the regulatory compliance scenario, it might be important for the Client Program to be able to determine whether it is safe **not** to store a copy of a given data object locally, i.e. whether or not the Data Provider can guarantee that it will provide the data at its associated, revisioned identifier at a later time. This way, the Client Program could avoid caching objects that it could simply retrieve again, and store only those objects whose Data Providers do not offer the required guarantee.

Use Case 5 (Query?) Get biological object's digital data of given type in given format

Primary Actor: Client Program on behalf of User
Scope: Client System, Data Provider System and IRSDS
Level: User Goal
The Client Program presents a biological object identifier (BO id) to the Resolution Stack along with BO ids for controlled vocabulary terms that describe the type and format of associated digital data that the User wishes to retrieve. The Resolution Stack presents these data to the Data Service, which returns the desired digital data object and associated identifier. If no such data object exists, the Data Service returns a fault, and the Resolution Stack reports an error to the Client Program.

Use Case 6 (Query?) Get data object as of a given timestamp

Primary Actor: Client Program on behalf of User
Scope: Client System, Data Provider System and IRSDS
Level: User Goal
This is sort of a special case of Use Case 5, in that part of the associated "type and format" information that the Client Program presents to the Resolution Stack could include a specification of a timestamp "as of" which the Client Program would like to retrieve the data object. In order that the timestamp specification have the same format as the other specifications, this timestamp could be encoded using a special BO identifier (e.g., urn:lsid:anAuthority:aNamespace:timestamp:20050105T153001.333) whose revision is the timestamp.

Use Case 7 Provide identified data

Primary Actor: Data Provider
Scope: Data Provider System, IRSDS and the caDSR
Level: Summary

The Data Provider will provide software that wraps existing data sources so they are available via standard data grid services. The Data Provider will register their authority string within the IRSDS. The Data Provider will publish the object models for their data objects in the caDSR. The Data Provider will deploy a mechanism within the Data Provider System to provide identifiers for all previously existing and new data objects. Finally, the Data Provider will deploy software to deliver data and associated metadata in the correct XML format(s).

Use Case 8 Provide relationships between biological objects and their digital data

Primary Actor: Data Provider
Scope: Data Provider System
Level: Summary
The Data Provider will provide the infrastructure to return metadata describing the relationships between a biological object identifier and associated digital data identifiers. This includes selecting or adding new vocabulary terms describing the types and formats of available digital data and storing these relationships in some sort of data store. There's probably more to this, but I'll leave it at that for now.

Use Case 9 Provide minimal metadata

Primary Actor: Data Provider
Scope: Data Provider System
Level: Summary
All caBIG Data Services will be required to provide a minimal set of metadata for each data object. This minimal set will include provenance, [to be completed once we decide what minimal metadata means].

Use Case 10 Move Data Service to new host

Primary Actor: Data Provider
Scope: Data Provider System and the IRSDS
Level: Summary
The Data Provider deploys a redundant copy of the Data Provider System on the new host. The Data Provider registers the new host within the IRSDS. Once identifiers can be resolved at the new host, the Data Provider deregisters the old host within the IRSDS. Finally, the Data Provider retires the Data Provider System on the old host.

Use Case 11 Retire Data Service

Primary Actor: Data Provider
Scope: Data Provider System and the IRSDS
Level: Summary
The Data Provider registers a special Data Service provided on some host within the IRSDS for its authority. This special service responds with a "service retired" message to all requests for identifier resolution. This allows clients to distinguish between a service that is currently unavailable but may become available later on and a service that will never be available again.